

Listing and Amendments to the Claims

This listing of claims will replace the claims that were published in the PCT Application and the International Preliminary Examination Report:

1. (currently amended) A programmable data processing device comprising :
 - a loading engine (~~LE~~) for receiving portions of code of a first type and/or data from a stream (~~DC~~) of a broadcast network in which said portions are repeatedly transmitted,
 - a storage means (~~C~~) for storing the portions received by the loading engine,
 - an execution engine (~~EE~~) for executing an application embodied by the received portions,
 - a translating engine (~~TE~~) for translating the first type code into a native code of the execution engine (~~EE~~)
 - ~~characterised in that~~ wherein the translating engine (~~TE~~) is adapted to store the thus compiled portion in the storage means (~~C~~), to compile at least a certain one of said received portions into native code, these portions being selected by a control information received from the stream (~~DC~~), and to interpret other portions of code, and that the execution engine (~~EE~~) is adapted to process compiled code and interpreted code within a same application.
2. (currently amended) The data processing device according to claim 1, wherein the stream (~~DC~~) is a DSM-CC carousel.
3. (currently amended) The data processing device according to claim 2, wherein said portion is all or a fraction of a DSM-CC module (~~T~~).
4. (original) The data processing device according to claim 3, wherein the control information is DSM-CC pre-fetch signalling.
5. (currently amended) The data processing device according to claim 3, wherein the translating engine (~~TE~~) is adapted to extract the control information from a payload module (~~T~~) of the DSM-CC carousel (~~DC~~).

6. (currently amended) The data processing device according to claim 5, wherein the translating engine (~~TE~~) is adapted to extract compiling optimization information relating to a portion of code to be compiled from said payload module (~~T~~) and to heed the compiling optimization in the process of compiling said portion of code.

7. (currently amended) The data processing device according to ~~one of claims 1 to 6~~ claim 1, wherein the translating engine (~~TE~~) is adapted to decide whether to compile or to interpret a given portion of first type code according to control information received from the execution engine (~~EE~~).

8. (currently amended) The data processing device according to claim 7, wherein the translating engine (~~TE~~), during compilation of a given first type code portion, is adapted to ignore control information requiring said portion to be interpreted, and to finish compiling the portion.

9. (currently amended) The data processing device according to claim 8, wherein the translating engine (~~TE~~), when receiving control information requiring a given first type code portion to be interpreted during compilation of said portion, is adapted to abandon the compilation and to start interpreting the portion.

10. (currently amended) A data processing method, comprising the steps of:
- a) receiving (~~a1, a1'~~) portions of code (~~I~~) of a first type and/ or data (~~D~~) from a stream (~~DC~~) of a broadcast network in which said portions (~~I, D~~) are repeatedly transmitted, wherein the set of portions transmitted in said stream (~~DC~~) embodies one or more data processing applications

- b) storing (~~a6, a9', b3~~) predetermined ones of said portions in a storage means (C),

- c) compiling in a translation engine at least one of said portions comprising first type code into native code of an execution engine, the compiled portions being selected by a control information received from the stream (~~DC~~),

- d) in the execution engine (~~EE~~), carrying out one of said data processing applications by executing (~~e6~~) the compiled native code (~~N~~) of the selected portions

belonging to said one application and by interpreting ~~(e7)~~ non-selected portions of this application.

11. (original) The data processing method of claim 10, comprising, between steps c and d, the step of receiving an instruction from a user specifying the application to be carried out in step d.

12. (currently amended) The data processing method according to ~~claims 10 or 11~~ claim 10, wherein the stream ~~(DC)~~ is a DSM-CC carousel.

13. (original) The data processing method according to claim 12, wherein said portion is all or a fraction of a DSM-CC module.

14. (original) The data processing method according to claim 13, wherein the control information is DSM-CC pre-fetch information.

15. (currently amended) The data processing method according to claim 13, wherein the control information is a payload module ~~(T)~~ of the DSM-CC carousel ~~(DC)~~.

16. (original) The data processing method of claim 15 wherein the control information further comprises compiling optimization information relating to a portion of code to be compiled, and the translation engine heeds the compiling optimization information when compiling said portion of code.

17. (currently amended) The data processing method according to ~~one of claims claim 10 to 13~~ claim 10, wherein the translating engine ~~(TE)~~ decides based on said control information from the execution engine ~~(EE)~~ whether to compile or to interpret a given first type code portion.

18. (currently amended) The data processing method according to claim 17, wherein if the translation engine ~~(TE)~~ receives control information requiring a given portion to be interpreted during compilation of said portion, it ignores ~~(e11)~~ the control information and finishes compiling the portion.

19. (currently amended) The data processing method according to claim 17, wherein if the translation engine (~~TE~~) receives control information requiring a given portion to be interpreted during compilation of said portion, it abandons the compilation (~~e11'~~) and starts interpreting the portion.

20. (currently amended) The data processing method according ~~one of claims 10 to 19~~ to claim 10 in which, after step c), memory space allocated to the first type code of the compiled portion is released for overwriting.